

CLAIMS

We claim:

1. A method for dynamically creating and maintaining a set of indices in a computer, wherein the indices identify a plurality of filters defining a network policy and wherein the indices are used by a firewall to identify a matching filter, comprising:
 - creating a first index conforming to a first index type;
 - identifying, in the first index, a first set of filters, each filter in the first set of filters specifying network packets subject to the network policy;
 - maintaining statistics including a selected criteria and a corresponding value, wherein the value identifies a number of filters from the first set of filters meeting the selected criteria;
 - determining that the corresponding value exceeds a threshold value;
 - creating a second index conforming to a second index type;
 - identifying, in the second index, a second set of filters, wherein the second set of filters are a subset of the first set of filters; and
 - removing identification of the subset of filters from the first index.
2. The method of claim 1, wherein the second index type is a linked list.
3. The method of claim 1, wherein the second index type is a tree data structure.
4. The method of claim 3, wherein the tree data structure is a single lookup tree.
5. The method of claim 3, wherein the tree data structure is a multiple lookup tree.
6. The method of claim 1, wherein the second index is a hash table.
7. The method of claim 1, wherein the plurality of filters include a set of filter conditions including a plurality of field types and corresponding field data, further comprising:

selecting one or more field types from the plurality of field types to be indexed.

8. The method of claim 1, wherein the second index is a linked list, and each filter includes a weight value, further comprising:

ordering the filters in the linked list such that a filter with a highest weight value is first in the linked list and a filter with the lowest weight value is last in the linked list.

9. The method of claim 1 further comprising:

adding a new filter to the firewall;
selecting an index from the first and second index, and
adding the new filter to the selected index.

10. The method of claim 1, wherein the second set of filters include filter conditions that meet the selected criteria.

11. A method for creating a filter index used to identify a plurality of filters used with a network firewall, each filter of the plurality of filters including a set of filter conditions and a filter weight, each filter condition including an individual field weight, comprising:

identifying an index type based upon the filter conditions of the plurality of filters;
identifying a subset of filter conditions to include in the index based upon an average field weight calculated from the individual field weight; and
selecting an order by which the subset of filter conditions are placed in the index.

12. The method of claim 11, wherein the index is a tree structure.

13. The method of claim 12, wherein the tree structure is a multi-lookup tree.

14. The method of claim 12, wherein the tree structure is a single lookup tree.

15. The method of claim 11, wherein the index is a hash table index.

16. A computer-readable medium for executing computer-readable instructions for dynamically creating and maintaining a set of indices in a computer, wherein the indices identify a plurality of filters defining a network policy and wherein the indices are used by a firewall to identify a matching filter, comprising:

creating a first index conforming to a first index type;

identifying, in the first index, a first set of filters, each filter in the first set of filters specifying network packets subject to the network policy;

maintaining statistics including a selected criteria and a corresponding value, wherein the value identifies a number of filters from the first set of filters meeting the selected criteria;

determining that the corresponding value exceeds a threshold value;

creating a second index conforming to a second index type;

identifying, in the second index, a second set of filters, wherein the second set of filters are a subset of the first set of filters; and

removing identification of the subset of filters from the first index.

17. The computer-readable medium of claim 16, wherein the plurality of filters include a set of filter conditions including a plurality of field types and corresponding field data, further comprising:

selecting one or more field types from the plurality of field types to be indexed.

18. The computer-readable medium of claim 16, wherein the index is a linked list, and each filter includes a weight value, further comprising:

ordering the filters in the linked list such that a filter with a highest weight value is first in the linked list and a filter with the lowest weight value is last in the linked list.

19. The computer-readable medium of claim 16, further comprising:

adding a new filter to the firewall;

selecting an index from the first and second index, and

adding the new filter to the selected index.

20. The computer-readable medium of claim 16 wherein the second set of filters include filter conditions that meet the selected criteria.

21. A computer-readable medium for executing computer-readable instructions for creating a filter index used to identify a plurality of filters used with a network firewall, each filter of the plurality of filters including a set of filter conditions and a filter weight, each filter condition including an individual field weight, comprising:

- identifying an index type based upon the filter conditions of the plurality of filters;
- identifying a subset of filter conditions to include in the index based upon an average field weight calculated from the individual field weight; and
- selecting an order by which the subset of filter conditions are placed in the index.

22. The method of claim 21, wherein the second index type is a linked list.

23. The method of claim 21, wherein the second index type is a tree data structure.

24. The method of claim 23, wherein the tree data structure is a single lookup tree.

25. The method of claim 23, wherein the tree data structure is a multiple lookup tree.

26. The method of claim 23, wherein the second index is a hash table.